

# 대한민국 특허청

## KOREAN INTELLECTUAL PROPERTY OFFICE

별첨 사본은 아래 출원의 원본과 동일함을 증명함.

This is to certify that the following application annexed hereto  
is a true copy from the records of the Korean Intellectual  
Property Office.

출원번호 : 10-2002-0058752  
Application Number PATENT-2002-0058752

출원년월일 : 2002년 09월 27일  
Date of Application SEP 27, 2002

출원인 : 삼성전자 주식회사  
Applicant(s) SAMSUNG ELECTRONICS CO., LTD.



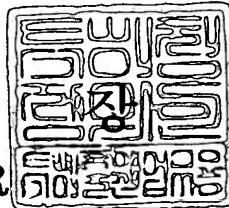
2002 년 11 월 09 일

특

허

청

COMMISSIONER



## 【서지사항】

**【서류명】** 특허출원서  
**【권리구분】** 특허  
**【수신처】** 특허청장  
**【참조번호】** 0002  
**【제출일자】** 2002.09.27  
**【국제특허분류】** H03M  
**【발명의 명칭】** 벡터 양자화를 위한 고속 탐색방법  
**【발명의 영문명칭】** FAST SEARCH METHOD FOR NEAREST NEIGHBOR VECTOR QUANTIZER

## 【출원인】

**【명칭】** 삼성전자 주식회사  
**【출원인코드】** 1-1998-104271-3

## 【대리인】

**【성명】** 이건주  
**【대리인코드】** 9-1998-000339-8  
**【포괄위임등록번호】** 1999-006038-0

## 【발명자】

**【성명의 국문표기】** 이남일  
**【성명의 영문표기】** LEE,Nam II  
**【주민등록번호】** 731116-1010020  
**【우편번호】** 442-718  
**【주소】** 경기도 수원시 팔달구 매탄4동 삼성1차아파트 1동 907호  
**【국적】** KR

## 【발명자】

**【성명의 국문표기】** 김용석  
**【성명의 영문표기】** KIM,Yong Serk  
**【주민등록번호】** 590820-1025416  
**【우편번호】** 442-713  
**【주소】** 경기도 수원시 팔달구 매탄4동 현대아파트 102동 405호  
**【국적】** KR

## 【발명자】

**【성명의 국문표기】** 황성규  
**【성명의 영문표기】** HWANG,Seong Kyu

【주민등록번호】	650820-1639343
【우편번호】	135-778
【주소】	서울특별시 강남구 대치2동 은마아파트 1동 806호
【국적】	KR
【발명자】	
【성명의 국문표기】	강상원
【성명의 영문표기】	KANG,Sang Won
【주민등록번호】	580320-1671318
【우편번호】	431-080
【주소】	경기도 안양시 동안구 호계동 1054-6 목련아파트 606동 803호
【국적】	KR
【발명자】	
【성명의 국문표기】	지상현
【성명의 영문표기】	CHI,Sang Hyun
【주민등록번호】	740319-1320923
【우편번호】	250-809
【주소】	강원도 홍천군 홍천읍 회망3리 4반 302-20
【국적】	KR
【심사청구】	청구
【취지】	특허법 제42조의 규정에 의한 출원, 특허법 제60조의 규정에 의한 출원심사를 청구합니다. 대리인 이건주 (인)
【수수료】	
【기본출원료】	20 면 29,000 원
【가산출원료】	13 면 13,000 원
【우선권주장료】	0 건 0 원
【심사청구료】	5 항 269,000 원
【합계】	311,000 원

**【요약서】****【요약】**

본 발명은 벡터 양자화를 위한 최적의 코드워드를 고속 탐색하는 방법에 관한 것이다. 이를 위해 본 발명에서는 입력 벡터의 지정된 한 요소의 왜곡(distortion)과 실험적으로 결정된 임계치(threshold)를 이용하여 코드-북에 있어 최적 코드 워드가 존재할 상위 및 하위 경계 값을 구한다. 그리고, 코드 워드의 특정한 하나의 요소의 크기에 따라 내림차순으로 재 정렬된 코드-북으로부터 코드북 탐색의 시작점과 끝점을 2진(binary) 탐색 방법을 사용하여 결정하고, 상기 결정한 시작점과 끝점에 의해 구하여진 탐색 범위 내에서만 풀-탐색 방법을 적용하여 벡터 양자화를 위한 최적 코드워드를 결정한다.

**【대표도】**

도 4

**【색인어】**

벡터 양자화, 코드워드, 풀-탐색, 고속 탐색

**【명세서】****【발명의 명칭】**

벡터 양자화를 위한 고속 탐색방법{FAST SEARCH METHOD FOR NEAREST NEIGHBOR VECTOR QUANTIZER}

**【도면의 간단한 설명】**

도 1은 종래 풀-탐색에 의해 벡터 양자화에 따른 최적 코드워드를 결정하는 제어 흐름을 보이고 있는 도면.

도 2는 종래 부분 거리 탐색에 의해 벡터 양자화에 따른 최적의 코드워드를 결정하는 제어 흐름을 보이고 있는 도면.

도 3은 종래 코드워드들 각각에 대응한 임계치에 의해 벡터 양자화에 따른 최적의 코드워드를 결정하는 제어 흐름을 보이고 있는 도면.

도 4는 본 발명의 실시 예에 따른 벡터 양자화를 위한 최적의 코드워드를 탐색하는 제어 흐름을 보이고 있는 도면.

도 5는 본 발명의 실시 예에 따른 코드-북의 구조를 도시하고 있는 도면.

도 6a와 도 6b는 본 발명의 실시 예에 따른 동작을 설명하기 위한 코드-북의 예를 도시하고 있는 도면.

**【발명의 상세한 설명】****【발명의 목적】****【발명이 속하는 기술분야 및 그 분야의 종래기술】**

- <7> 본 발명은 벡터 양자화 방법에 관한 것으로, 특히 벡터 양자화를 위한 최적의 코드워드를 고속 탐색하는 방법에 관한 것이다.
- <8> 통상적으로 벡터 양자화(Vector Quantization)는 주어진 비트율에서 PCM, DPCM, ADPCM과 같은 스칼라 양자화(Scalar Quantization)보다 작은 양자화 에러를 제공함으로써, 데이터 압축(Data Compression) 및 인식 분야에서 가장 널리 사용되어온 양자화 방법이다. 즉, 상기 벡터 양자화는 음성 코딩(Speech coding), 오디오 코딩(Audio Coding), 이미지 코딩(Image Coding), 음성 인식(Speech Recognition) 등 광범위한 분야에 응용되고 있다.
- <9> 한편, 벡터 양자화를 위해서는 복수개의 코드워드들로 이루어진 코드북을 탐색하고, 상기 탐색을 통해 입력 벡터를 양자화함에 있어 최적의 코드워드를 찾는 동작이 반드시 요구된다. 상기 최적의 코드워드는 상기 입력 벡터와의 양자화 에러가 최소로 발생하는 코드워드를 지칭한다. 종래 상기 코드-북의 탐색을 통해 최적의 코드워드를 찾기 위해 제안되고 있는 방법들은 크게 세 가지로 구분할 수 있다. 첫 번째 방법은 코드-북을 구성하는 모든 코드 워드들에 대해 벡터 양자화를 수행하는 풀-탐색에 의한 벡터 양자화 방법이고, 두 번째 방법은 부분 거리 탐색(Partial Distance Search) 방식을 이용한 벡터 양자화 방법이며, 마지막으로 세 번째 방법은 임계치를 이용한 벡터 양자화 방법이다.

- <10> 먼저, 첫 번째 방법에 해당하는 풀-탐색에 의해 벡터 양자화를 수행하는 방법에 대해 살펴보면 다음과 같다.
- <11> 도 1은 상기 풀-탐색(full-searcher)에 의한 벡터 양자화(NNVQ : Nearest Neighbor Vector Quantization)를 위한 제어 흐름을 보이고 있는 도면이며, 이를 참조하여 상기 풀-탐색에 의한 벡터 양자화 과정을 설명하면 다음과 같다.
- <12> 110단계에서 벡터 양자화를 위한 각 파라미터 값들( $D_{\min}$ ,  $i$ ,  $\text{index}$ )을 초기화한다. 즉, 상기  $D_{\min}$ 은 max로 설정하고, 상기  $i$ 와 상기  $\text{index}$ 는 0을 설정한다. 상기  $D_{\min}$ 은 코드-북을 구성하는 코드워드들( $\overline{y_i}$ )과 입력 벡터( $\overline{x}$ )간의 양자화 에러들 중 가장 작은 값을 설정하기 위한 파라미터이다. 상기  $i$ 는 상기 입력 벡터( $\overline{x}$ )와의 양자화 에러를 계산하기 위해 상기 코드워드들( $\overline{y_i}$ ) 중 하나의 코드워드를 선택하기 위한 파라미터이다. 상기 코드-북을 구성하는 상기 코드워드들의 총 수가  $N$ 개라 가정할 시 상기  $i$ 는 1부터  $N$ 의 범위를 가진다. 상기  $\text{index}$ 는 상기 입력 벡터( $\overline{x}$ )와 가장 작은 양자화 에러 값( $D_{\min}$ )에 대응하는 코드워드를 가리키는 인덱스를 의미한다. 따라서, 상기  $\text{index}$ 는 상기  $i$ 의 범위인 1부터  $N$ 까지의 정수들 중 임의의 값이 될 것이다.
- <13> 상기 파라미터들의 초기화가 완료되면 112단계로 진행하여 상기  $i$ 가 상기  $N$ 과 같은지를 비교함으로써 상기 모든 코드워드들에 대한 양자화 에러 계산이 완료되었는지를 판단한다. 상기 모든 코드워드들에 대한 양자화 에러 계산이 완료되지 않았다고 판단되면 114단계로 진행한다. 상기 양자화 에러 계산은 양자화 에러를 최소로 하는 최적의 코드워드를 결정하기 위한 절차이다. 상기 114단계에서는 상기  $i$ 를 1 증가시킴으로써 다음 인덱스의 코드워드가 선택될 수 있도록 한다. 상기 114단계에서 다음 인덱스의 코드워드가 선택되면 116단계로 진행하여 상기 선택된 코드워드와 입력 벡터간의 양자화 에러  $D_i$

를 계산한다. 일 예로서 차원이 k인 입력 벡터  $\bar{x}$ 와 i번째 코드 워드  $y_i$ 간의 양자화 에러  $D_i$ 는 하기 <수학식 1>로서 계산되어질 수 있다.

<14>      **【수학식 1】**       $d[\bar{x}, y_i] = D_i = \frac{1}{k} (\bar{x} - y_i)(\bar{x} - y_i)^T = \frac{1}{k} \sum_{j=1}^k [\bar{x}_j - y_{ij}]^2$

<15>      여기서, k는 입력 벡터  $\bar{x}$ 의 차수를 의미하며, 모든  $\{y\}_i$ 는 코드 워드들 중 i번째 코드 워드를 의미한다. 또한  $\bar{x}_j$ ,  $y_{ij}$ 는 각각 벡터  $\bar{x}$ ,  $y_i$ 의 j번째 요소를 의미한다.

<16>      상기 116단계에 의해 현재 i에 대응하는  $D_i$ 의 계산이 완료되면 118단계로 진행하여 상기  $D_i$ 가 상기  $D_{min}$ 보다 큰지를 판단한다. 만약, 상기  $D_i$ 가 상기  $D_{min}$ 보다 크다면 상기  $D_{min}$ 을 변경하지 않고, 상기 112단계로 진행하여 전술한 122단계 내지 116단계를 수행한다. 하지만, 상기  $D_i$ 가 상기  $D_{min}$ 보다 크지 않으면 120단계로 진행하여 상기  $D_i$ 를 상기  $D_{min}$ 으로 변경하고, index를 i로 설정한 후 상기 112단계로 진행한다. 상기  $D_i$ 를 상기  $D_{min}$ 으로 설정하는 것은 상기  $D_{min}$ 을 항상 최소의 값으로 유지하기 위함이며, 이때 i를 index로 설정함으로써 해당 코드 워드를 지정할 수 있도록 한다.

<17>      한편, 상기 114단계 내지 120단계는 상기 112단계에서 N개의 모든 코드 워드들을 대상으로 하여 벡터 양자화를 위한 최적의 코드 워드의 결정이 완료되었다고 판단될 때까지 반복적으로 수행된다. 이와 같은 동작에 의해, 상기 모든 코드 워드들을 대상으로 하여 벡터 양자화를 위한 최적의 코드 워드를 결정하는 과정이 완료되면 상기  $D_{min}$ 은 최소의 양자화 에러 값을 가지며, index는 이에 대응한 코드 워드를 지정하게 될 것이다. 따라서, 122단계에서는 상기 index로 설정된 값을 최상의 인덱스(best\_index)로서 설정



하게 되며, 상기 입력 벡터  $\bar{x}$ 는 상기 최상의 인덱스(best\_index)에 대응하는 코드 워드  $y_{index}$ 에 의해 양자화가 이루어지게 된다.

<18> 전술한 첫 번째 방법의 경우에는 코드-북을 구성하는 모든 코드워드들 각각에 대하여 입력 벡터와의 양자화 에러  $D_i$ 를 계산하여야 한다. 이러한 상기 첫 번째 방법은 두 가지의 근본적인 문제점을 가진다. 그 하나는 상기 입력 벡터의 차원  $k$ 가 증가함에 따라 상기 코드-북 탐색에 필요한 연산의 복잡도가 기하급수적으로 증가하게 되는 것으로서, 이러한 문제는 벡터 양자화를 실시간으로 처리하고자 할 경우 가장 먼저 해결되어야 할 문제이다. 다른 하나는 상기 입력 벡터의 차원  $k$ 가 증가함에 따라 코드 워드들을 저장하기 위해 요구되는 메모리의 크기가 기하급수적으로 증가한다는 것이다.

<19> 예컨대, 벡터의 차원이  $k$ 이고, 코드-북의 크기가  $N$ 일 경우 전술한 풀-탐색 방식을 적용한다면, 입력 벡터를 양자화하기 위하여  $k$ 개의 곱셈과  $(2k-N)$ 개의 가산과 감산 및 비교 연산이 필요하다. 이와 같이 전술한 풀-탐색 방식의 경우에는 코드-북의 크기에 비례하며, 상기 코드-북을 저장하기 위한 메모리는 벡터의 차원  $k$ 가 증가함에 따라 기하급수적으로 증가하게 된다.

<20> 다음으로, 두 번째 방법에 해당하는 부분 거리 탐색(PDS : Partial Distance Search)을 이용한 벡터 양자화 방법에 대해 살펴보면 다음과 같다. 상기 부분 거리 탐색 방식을 이용한 벡터 양자화 방법은 상기 풀-탐색 방식에서 요구되는 많은 계산량을 줄이기 위해 IEEE(vol.COM-33, p1132~p1133)를 통해 제안되었다. 상기 IEEE(vol.COM-33, p1132~p1133)를 통해 제안되고 있는 두 번째 방법은 입력 벡터와 각 후보 코드 워드와의 양자화 에러 계산 시, 후보 벡터의 모든 요소에 대해 계산하지 않고  $r$  개의 요소에

대해 에러를 계산한다. 여기서  $r = k$ 이다. 즉, 복수의 코드워드들 중 임의의 코드워드에 대한 양자화 에러를 계산하는 중에 임의의 순간  $r$ 에서 이전까지의 최소 양자화 에러 값  $D_{\min}$ 보다 크면 상기 임의의 코드워드에 대한 더 이상의 양자화 에러를 계산하더라도 상기  $D_{\min}$ 보다 큰 값일 것임은 자명하기 때문이다.

<21> 도 2는 상기 부분 거리 탐색을 이용한 벡터 양자화를 위한 제어 흐름을 보이고 있는 도면이며, 이를 참조하여 상기 부분 거리 탐색에 의한 벡터 양자화 과정을 설명하면 다음과 같다.

<22> 210단계에서 벡터 양자화를 위한 각 파라미터 값들( $D_{\min}$ ,  $i$ ,  $index$ )을 초기화한다. 즉, 상기  $D_{\min}$ 은 max로 설정하고, 상기  $i$ 와 상기  $index$ 는 0을 설정한다. 상기 파라미터들의 초기화가 완료되면 212단계로 진행하여 임의의 코드워드  $y_i$ 에 대해 양자화 에러 계산시 사용된  $y_i$ 의 요소수를 카운트하기 위한  $r$ 을 0으로 설정한다. 그리고, 214단계로 진행하여 상기  $i$ 가 상기  $N$ 과 같은지를 비교함으로서 상기 모든 코드워드들에 대한 양자화 에러 계산이 완료되었는지를 판단한다. 상기 모든 코드워드들에 대한 양자화 에러 계산이 완료되었다고 판단되면 228단계로 진행한다. 상기 216단계에서는 상기  $i$ 를 1 증가시킴으로서 다음 인덱스의 코드워드가 선택될 수 있도록 한다. 상기 216단계에서 다음 인덱스의 코드워드가 선택되면 218단계로 진행하여 상기  $r$ 을 1 증가시킴으로서 해당 코드워드의 요소들 중 다음 요소가 선택될 수 있도록 한다. 상기  $i$ 와 상기  $r$ 이 결정되면 220단계로 진행하여 상기 선택된 코드워드의 처음 요소로부터 상기  $r$ 에 의해 선택된 요소까지와 입력 벡터간의 양자화 에러  $D_r$ 을 계산한다. 일 예로서 상기  $r$ 의 범위에서 차원이  $k$ 인 입력 벡터  $\vec{x}$ 와  $i$ 번째 코드 워드  $y_i$ 간의 양자화 에러  $D_r$ 은 하기 <수학식 2>로서 계산되어질 수 있다.

<23> **【수학식 2】** 
$$D_r = \sum_{j=1}^r [\bar{x}_j - y_{ij}]^2$$

<24> 여기서,  $\bar{x}_j$ 는 입력 벡터  $\bar{x}$ 의 j번째 요소이며,  $y_{ij}$ 는 i번째 코드워드  $y_i$ 의 j번째 요소를 의미한다.

<25> 상기 220단계에 의해 임의의 코드워드  $y_i$ 의 r번째 요소까지에 대응하는  $D_r$ 의 계산이 완료되면 222단계로 진행하여 상기  $D_r$ 이 상기  $D_{min}$ 보다 큰지를 판단한다. 상기  $D_{min}$ 은 이전 코드워드  $y_{i-1}$ 까지의 모든 코드워드들에 의해 계산되어진 양자화 에러 값들 중 최소 값이 될 수 있다. 만약, 상기  $D_r$ 이 상기  $D_{min}$ 보다 크다면 상기  $D_{min}$ 을 변경하지 않고, 상기 212단계로 진행한다. 이는 상기 임의의 코드워드  $y_i$ 에 대한 양자화 에러 계산을 중단하고, 다음 코드워드  $y_{i+1}$ 에 대한 양자화 에러 계산을 수행하도록 하기 위함이다. 하지만, 상기 222단계에서  $D_r$ 이 상기  $D_{min}$ 보다 크지 않으면 224단계로 진행하여 상기 r이 상기 임의의 코드워드  $y_i$ 의 차원인 k와 일치( $r=k$ )하는 지를 판단한다. 즉, 상기  $D_r$ 이 상기 임의의 코드워드  $y_i$ 를 구성하는 모든 요소들에 대해 계산되어진 양자화 에러 값인지를 판단하게 된다. 상기 224단계에서 모든 요소들에 대한 양자화 에러 계산이 이루어지지 않았다고 판단되면 상기 218단계로 진행하며, 그렇지 않으면 226단계로 진행한다. 상기 218단계로 진행하면 상기 r을 1 증가시킴으로서 상기 임의의 코드워드  $y_i$ 를 구성하는 요소들 중 현재까지 양자화 에러 계산이 이루어진 요소를 포함한 다음 요소까지를 양자화 에러 계산에 감안하기 위함이다. 그 후 220단계를 통해 양자화 에러  $D_r$ 의 계산이 이루어지게 된다. 전술한 바와 같이 상기 임의의 코드워드  $y_i$ 를 구성하는 모든 요소들 또는 계산된 양자화 에러 값  $D_r$ 이  $D_{min}$ 보다 커 질 때까지 상기 임의의 코드워드  $y_i$ 에 대한 양자화 에러 계산 과정을 반복적으로 수행된다.

<26> 하지만, 상기  $r$ 과  $k$ 가 일치함에 따라 상기 226단계로 진행하면 상기  $D_r$ 을 상기  $D_{\min}$ 으로 변경하고,  $index$ 를  $i$ 로 설정한 후 상기 212단계로 진행하여 다음 코드워드에 대응한 양자화 에러 값 계산 절차를 수행한다. 상기  $D_r$ 을 상기  $D_{\min}$ 으로 설정하는 것은 상기  $D_{\min}$ 을 항상 최소의 값으로 유지하기 위함이며, 이때  $i$ 를  $index$ 로 설정함으로써 해당 코드 워드를 지정할 수 있도록 한다. 즉, 상기 226단계에서  $D_r$ 을  $D_{\min}$ 으로 설정하는 이유는 상기 222단계에서  $D_r$ 이  $D_{\min}$ 보다 크지 않다고 판단되었으므로 현재까지의 양자화 에러 값들 중 상기  $y_i$ 를 구성하는 모든 요소들에 의해 계산된 양자화 에러 값이 가장 작은 것으로 판단할 수 있기 때문이다.

<27> 한편, 상기 212단계 내지 226단계는 상기 214단계에서  $N$ 개의 모든 코드 워드들에 대응한 벡터 양자화를 위한 최적의 코드 워드의 결정이 완료되었다고 판단될 때까지 반복적으로 수행된다. 이와 같은 동작에 의해, 상기 모든 코드 워드들을 대상으로 하여 벡터 양자화를 위한 최적의 코드 워드를 결정하는 과정이 완료되면 상기  $D_{\min}$ 은 최소의 양자화 에러 값을 가지며,  $index$ 는 이에 대응한 코드 워드를 지정하게 될 것이다. 따라서, 228단계에서는 상기  $index$ 로 설정된 값을 최상의 인덱스(best\_index)로서 설정하게 되며, 상기 입력 벡터  $\bar{x}$ 는 상기 최상의 인덱스(best\_index)에 대응하는 코드 워드  $y_{index}$ 에 의해 양자화가 이루어지게 된다.

<28> 전술한 두 번째 방법의 경우에는 첫 번째 방법에 비해 많은 량의 곱셈 계산 량을 줄일 수 있으나 추가적인 비교 연산이 요구되는 다른 문제점을 가진다.

<29> 마지막으로, 세 번째 방법에 해당하는 이용한 벡터 양자화 방법에 대해 살펴보면 다음과 같다. 상기 임계치를 이용한 벡터 양자화 방법은 'Erik Staats'에 의해 출원된 미국 출원번호 US5,619,717(1997년 4월 8일)에서 개시되고 있다. 상기 세 번째 방법은

코드워드들간의 거리를 미리 계산하여 놓은 표를 이용하여 벡터 양자화를 수행함으로써 계산량을 줄이는 방식을 채택하고 있다. 즉, 코드-북을 구성하는 코드워드들을 양자화 빈도 수 또는 코드워드간의 거리를 기준으로 코드-북의 코드워드들을 재배열하고, 상기 재배열된 코드워드들 각각에 관련된 임계치들을 별도로 저장함으로서 입력 벡터와 상응하는 임계치의 코드워드에 의해 양자화를 수행하는 방법을 제안한다.

<30> 도 3은 미리 계산되어진 코드워드들간의 거리를 이용하여 벡터 양자화를 수행하기 위한 제어 흐름을 보이고 있는 도면이며, 이를 참조하여 미리 저장된 코드워드들간의 거리에 의한 벡터 양자화 과정을 설명하면 다음과 같다.

<31> 310단계에서 코드-북내의 코드 워드들을 양자화 빈도 수, 또는 코드 워드간의 거리를 기준으로 재배열한다. 그리고, 312단계로 진행하여 상기 재배열된 코드-북의 코드워드들에 대해 상기 코드워드들 각각에 관련된 임계치들을 생성한다. 예컨대, 상기 코드-북의 크기가 N이라 가정하면 상기 임계치들은 N 번째 코드워드를 제외한 나머지 코드워드들 각각에 대해 생성되어진다. 상기 임계치는 해당 코드워드와 거리가 가장 가까운 코드워드와의 거리의 1/2로서 정의될 수 있다. 하기 <수학식 3>에서는 j번째 코드워드  $y_j$ 에 대한 임계치  $Thresh_j$ 를 나타내고 있다.

<32> 【수학식 3】  $Thresh_j = 0.5 \times \min_{k > j} \|y_j - y_k\|$

<33> 전술한 바에 의해 구하여진 임계치들 각각은 해당 코드워드들에 대응하여 저장된다. 따라서, 제안된 세 번째 방법을 위해서는 상기 임계치들을 저장하기 위한 별도의 메모리 영역이 요구된다. 상기 임계치들의 생성이 완료되면 314단계로 진행하여 입력벡터  $\bar{x}$ 를 상기 재배열된 코드-북과 상기 저장된 임계치들을 이용하여 양자화를 수행한다. 만약

, 입력벡터  $\bar{x}$ 와 코드워드  $y_j$ 사이의 거리가 상기 코드워드  $y_j$ 에 상응하는 임계치  $\text{Thresh}_j$ 보다 작을 경우, 가장 가까운 코드워드를 탐색하는 과정은 종료된다. 그리고, 상기 입력벡터  $\bar{x}$ 는 코드워드  $y_j$ 에 의해 양자화를 수행한다.

- <34> 전술한 세 번째 방법의 경우에는 코드워드들을 저장하기 위한 메모리 공간과 별도로 미리 계산된 코드워드들간의 거리를 저장하기 위한 메모리 공간을 추가로 확보하여야 한다. 그로 인해, 전술한 종래 벡터 양자화 방법들에 비해 많은 메모리 공간이 요구되는 단점이 있다.

【발명이 이루고자 하는 기술적 과제】

- <35> 따라서, 상기한 바와 같은 문제점들을 해결하기 위한 본 발명의 목적은 풀-탐색 대상인 코드-북을 구성하는 코드 워드들의 탐색 범위를 줄여 벡터 양자화를 수행하는 방법을 제공함에 있다.
- <36> 본 발명의 다른 목적은 성능 감쇠 없이 적은 계산 량으로 코드-북을 탐색하는 벡터 양자화 방법을 제공함에 있다.
- <37> 본 발명의 또 다른 목적은 추가적인 메모리의 증가 없이 적은 계산 량으로 코드-북을 탐색하는 방법을 제공함에 있다.
- <38> 상기한 목적들을 달성하기 위한 제1견지에 있어, 본 발명은 입력 벡터를 벡터 양자화하기 위한 최적의 코드워드를 복수의 코드워드들로 구성된 코드-북으로부터 탐색하는 방법에 있어서, 상기 입력 벡터를 구성하는 복수의 요소들 중 임의 위치에 대응하는 요소 값에 미리 결정된 임계치를 감산 및 가산하여 하위 경계 값과 상위 경계 값을 계산하

는 과정과, 상기 코드-북을 구성하는 상기 코드워드들 각각에 있어 상기 임의 위치에 대응하는 요소 값들 중 상기 하위 경계 값과 상기 상위 경계 값 사이에 존재하는 요소 값들을 가지는 코드워드들을 탐색 대상 코드워드들로 지정하는 과정과, 상기 탐색 대상 코드워드들을 대상으로 하여 상기 입력 벡터와의 양자화 에러 값을 계산하는 풀-탐색을 통해 최소 양자화 에러 값을 가지는 최적의 코드워드들 결정하는 과정을 포함함을 특징으로 한다.

<39> 이를 위해 보다 바람직하기로는, 상기 탐색 대상 코드워드들을 지정하는 과정이, 상기 코드-북을 구성하는 상기 코드워드들 각각에 있어 상기 임의 위치에 대응하는 요소 값들의 크기에 의해 내림차순으로 상기 코드-북의 코드워드들을 재 정렬하는 과정과, 상기 상위 경계 값을 상기 재 정렬한 코드워드들 각각에 있어 상기 임의 위치에 대응하는 요소 값들과 비교하여 최초로 상기 상위 경계 값보다 작은 요소 값에 대응하는 시점을 시작점으로 결정하는 과정과, 상기 하위 경계 값을 상기 재 정렬한 코드워드들 각각에 있어 상기 임의 위치에 대응하는 요소 값들과 비교하여 최초로 상기 하위 경계 값보다 작은 요소 값에 대해 큰 요소 값에 대응하는 시점을 끝점으로 결정하는 과정과, 상기 시작점과 상기 끝점간에 존재하는 코드워드들을 상기 탐색 코드워드들로 지정하는 과정으로 이루어지는 것이다.

<40> 이를 위해 보다 바람직하기로는, 상기 탐색 대상 코드워드들을 지정하는 과정이, 상기 코드-북을 구성하는 상기 코드워드들 각각에 있어 상기 임의 위치에 대응하는 요소 값들의 크기에 의해 올림차순으로 상기 코드-북의 코드워드들을 재 정렬하는 과정과, 상기 하위 경계 값을 상기 재 정렬한 코드워드들 각각에 있어 상기 임의 위치에 대응하는 요소 값들과 비교하여 최초로 상기 하위 경계 값보다 큰 요소 값에 대응하는 시점을 시

작점으로 결정하는 과정과, 상기 상위 경계 값을 상기 재 정렬한 코드워드들 각각에 있어 상기 임의 위치에 대응하는 요소 값들과 비교하여 최초로 상기 하위 경계 값보다 큰 요소 값에 대해 작은 요소 값에 대응하는 시점을 끝점으로 결정하는 과정과, 상기 시작점과 상기 끝점간에 존재하는 코드워드들을 상기 탐색 코드워드들로 지정하는 과정으로 이루어지는 것이다.

#### 【발명의 구성 및 작용】

<41> 이하, 본 발명에 따른 바람직한 실시 예를 첨부한 도면을 참조하여 상세히 설명한다. 하기의 설명에서는 본 발명에 따른 동작을 이해하는데 필요한 부분만이 설명되며 그 이외 부분의 설명은 본 발명의 요지를 흐트리지 않도록 생략될 것이라는 것을 유의하여야 한다.

<42> 우선, 본 발명에서는 입력 벡터의 지정된 한 요소의 왜곡(distortion)과 실험적으로 결정된 임계치(threshold)를 이용하여 코드-북에 있어 최적 코드 워드가 존재할 상위 및 하위 경계 값을 구한다. 그리고, 코드 워드의 특정한 하나의 요소의 크기에 따라 내림차순으로 재 정렬된 코드-북으로부터 코드북 탐색의 시작점과 끝점을 2진(binary) 탐색 방법을 사용하여 결정하고, 상기 결정한 시작점과 끝점에 의해 구하여진 탐색 범위 내에서만 풀-탐색 방법에 의해 양자화를 위한 최적 코드워드를 결정한다. 여기서, 임계치는 충분한 길이의 훈련(training) 벡터들에 대한 최적 MSE(Mean Squared Error) 값들 중 최대 값을 이용하여 임계치를 결정하고, 임의의 입력 벡터와 최적 코드워드로 지정된 단일 요소 값간의 거리가 상기 임계치 보다 작다는 성질을 이용해서, 입력 벡터에 대한 최적 코드워드가 존재하는 범위를 결정한다. 한편, 상기 탐색범위의 결정에 사용되는 요



소의 선정은, 벡터 요소들 중에서 탐색범위의 감소율을 최대화시키는 요소를 실험적으로 미리 결정하도록 한다.

<43> 즉, 입력 벡터를 위한 최적의 코드워드를 본 발명의 실시 예를 구현하기 위해서는 훈련 벡터들에 대한 최적의 MSE 값들 중 최대 값을 결정하고, 상기 결정한 최대 값과 실험에 의해 결정된 최적의 제어 요소를 이용하여 임계치를 결정하여야 한다. 그리고, 벡터 요소들 중에서 탐색범위의 감소율을 최대화시키는 요소(이하  $j$ 번째 요소라 칭함)를 실험적으로 미리 결정하여 이 요소의 크기값에 따라 코드북을 재정렬 한다. 상기 임계치를 결정하면 입력 벡터를 구성하는 복수의 요소들 중  $j$ 번째 요소에 상기 결정한 임계치를 감산함으로서 하위 경계 값을 결정하고, 상기 입력 벡터를 구성하는 복수의 요소들 중  $j$ 번째 요소에 상기 결정한 임계치를 가산함으로서 상위 경계 값을 결정한다. 그 후 상기 코드-북을 구성하는 상기 코드워드들 각각에 있어 상기  $j$ 번째 요소 값들을 상기 하위 경계 값과 상기 상위 경계 값에 비교함으로서 상기 하위 경계 값과 상기 상위 경계 값 사이에 존재하는 요소 값들을 가지는 코드워드들을 탐색 대상 코드워드들로 지정한다. 상기 탐색 대상 코드워드들을 지정하면 상기 탐색 대상 코드워드들 각각과 상기 입력 벡터에 대해 풀-탐색 방식을 적용함으로서 양자화 에러 값들을 계산한다. 상기 계산한 양자화 에러 값들 중 최소의 양자화 에러 값을 구하고, 상기 구하여진 최소의 양자화 에러 값이 계산된 코드워드를 최적의 코드워드로서 결정한다. 여기서  $j$ 는 1 내지  $k$ 의 벡터 요소들 중에서 탐색범위의 감소율을 최대화시키는 요소의 순서로써 실험적으로 미리 결정된 값이다.

<44> 도 4는 본 발명의 실시 예에 따라 양자화를 수행하기 위한 코드워드를 결정하는 제어 흐름을 보이고 있는 도면이며, 도 5는 본 발명의 적용하기 위한 코드-북 구조의 일 예를 도시한 도면이다.

<45> 상기 도 4에서 의한 코드워드 결정은 코드-북을 구성하는 코드워드들에 있어 상위 경계 값과 하위 경계 값을 계산하는 제1과정과, 상기 코드-북의 코드워드들을 재 정렬하는 제2과정과, 상기 코드-북의 코드워드들 중 탐색 범위의 시작점과 끝점을 결정하는 과정과, 상기 결정한 탐색 범위내의 코드워드들에 대해 풀-탐색을 수행하는 과정으로 이루어진다.

<46> 상기 도 4를 참조하면, 410단계에서 상기 제1과정에서 상위 경계 값과 하위 경계 값을 계산하기 위한 왜곡 값을 훈련 벡터로부터 계산한다. 이때 훈련벡터는 양자화기의 입력벡터와 같은 특성을 가지는 충분히 많은 수의 벡터들로 정의되며, 상기 왜곡 값은 훈련 벡터들에 대한 최적의 MSE 값들 중 최대 값으로서 결정할 수 있는데, 이는 하기 <수학식 4>에 의해 계산할 수 있다.

<47> **【수학식 4】** 
$$\text{dist} = \max_{\text{for all } \bar{x}} \left\{ \min_{i=1, \dots, N} \|\bar{x} - y_i\|^2 \right\}$$

<48> 여기서, N은 코드-북의 크기,  $\bar{x}$ 는 입력 벡터,  $y_i$ 는 i번째 코드워드를 의미한다.

<49> 상기 410단계에서 왜곡 값들에 대한 계산이 완료되면 412단계로 진행하여 임계치 thrs를 결정한다. 상기 임계치 thrs는 상기 훈련 시퀀스로부터 실험적으로 결정된 상기 왜곡 값과 제어 요소인 P를 이용하여 계산된다. 이는 하기 <수학식 5>와 같이 나타낼 수 있다.

<50> **【수학식 5】** 
$$\text{thrs} = P \times \sqrt{\text{dist}}$$

<51> 여기서, P는 탐색범위의 크기와 MSE 성능을 조절하는 제어 요소이다. 상기 P는 0과 1 사이의 값을 가지며, 값이 클수록 상기 탐색 범위는 증가한다. 상기 P의 값이 크다는 것은 1에 가깝다는 것을 의미한다. 한편, 상기 P는 일반적인 풀-탐색과 동일한 성능을 나타내면서 탐색 범위를 최소화하도록 실험적으로 결정된다.

<52> 414단계에서는  $N \times k$  크기를 가지는 코드-북의 코드워드들( $y_1, y_2, y_3, y_4, \dots, y_{N-1}, y_N$ ) 각각에 있어 임의의 위치  $j$ 에 존재하는 요소들( $y_{1,j}, y_{2,j}, y_{3,j}, y_{4,j}, \dots, y_{N-1,j}, y_{N,j}$ )의 값들을 비교한다. 그리고, 상기 요소들의 값들( $y_{1,j}, y_{2,j}, y_{3,j}, y_{4,j}, \dots, y_{N-1,j}, y_{N,j}$ )의 크기에 따라 상기 코드-북의 코드워드들( $y_1, y_2, y_3, y_4, \dots, y_{N-1}, y_N$ )을 재 정렬한다. 상기 코드워드들( $y_1, y_2, y_3, y_4, \dots, y_{N-1}, y_N$ )을 재 정렬하는 방법은 오름차순 또는 내림차순에 관계없이 크기 순에 의해 정렬되면 된다.

<53> 상기 코드-북의 정렬이 완료되면 416단계로 진행하여 상기 <수학식 5>로부터 결정된 상기 임계치  $thrs$ 에 의해 상위 및 하위 경계값을 계산한다. 예컨대, 입력 벡터  $\bar{x}$ 의 지정된 한 요소 값( $j$  번째 요소)과 상기 412단계에서 실험에 의해 결정된  $thrs$ 를 이용하여 최적의 코드워드가 존재할 영역의 상위 경계 값과 하위 경계 값을 계산한다. 따라서, 상기 도 5에서 도시하고 있는 바와 같이  $N$ 개의 코드워드들( $y_1, y_2, y_3, y_4, \dots, y_{N-1}, y_N$ )과 상기 코드워드들 각각의 요소들이  $k$ 개로 이루어진 코드-북을 대상으로 할 때 상기 상위 경계 값과 상기 하위 경계 값은 하기 <수학식 6>으로 정의될 수 있다.

<54> 【수학식 6】  $|\bar{x}_j - y_{ij}| < thrs, (0 \leq i \leq N, 0 \leq j \leq k)$

<55> 상기 <수학식 6>은 하기 <수학식 7>로 변환될 수 있다.

<56> 【수학식 7】  $\bar{x}_j - thrs < y_{ij} < \bar{x}_j + thrs, (0 \leq i \leq N, 0 \leq j \leq k)$

- <57> 여기서,  $\overline{x}_j$ 는 입력 벡터  $\overline{x}$ 의  $j$  번째 요소이고,  $y_{ij}$ 는  $i$ 번째 코드워드의  $j$ 번째 요소( $y_{1,j}, y_{2,j}, y_{3,j}, y_{4,j}, \dots, y_{N-1,j}, y_{N,j}$ )를 의미한다. 한편,  $\overline{x}_j - \text{thrs}$ 는 하위 경계 값을 의미하며,  $\overline{x}_j + \text{thrs}$ 는 상위 경계 값을 의미한다.
- <58> 418단계에서는 2진 검색 방법을 이용하여 탐색 대상의 시작점과 끝점을 각각 결정한다. 상기 시작점과 상기 끝점은 상기 416단계에서 계산한 상위 경계 값과 하위 경계 값에 의해 산출할 수 있다. 예컨대, 상기 코드-북의 재 정렬을 오름차순에 의해 재 정렬하였을 경우에는 상기 하위 경계 값에 대해 2진 검색 방법을 적용함으로써 시작점을 결정하고, 상기 상위 경계 값에 대해 2진 검색 방법을 적용함으로써 끝점을 결정한다. 하지만, 상기 코드-북의 재 정렬을 내림차순에 의해 재 정렬하였을 경우에는 상기 상위 경계 값에 대해 2진 검색 방법을 적용함으로써 시작점을 결정하고, 상기 하위 경계 값에 대해 2진 검색 방법을 적용함으로써 끝점을 결정한다. 예컨대, 상기 내림차순에 의해 재 정렬이 이루어졌다고 가정할 시 상기 상위 경계 값에 의해 시작점을 결정하고, 상기 하위 경계 값에 의해 끝점을 결정하는 것은 다음과 같다. 먼저, 상기 재 정렬된 코드워드들 각각의  $j$ 번째 요소 값들을 상기 상위 경계 값과 비교하고, 상기  $j$ 번째 요소 값들 중 상기 상위 경계 값보다 작아지는 시점의 요소 값을 시작점으로 결정한다. 그리고, 상기 재 정렬된 코드워드들 각각의  $j$ 번째 요소 값들을 상기 하위 경계 값과 비교하고, 상기  $j$ 번째 요소 값들 중 상기 하위 경계 값보다 작아지는 시점의 요소 값을 끝점으로 결정한다. 따라서, 상기 시작점과 상기 끝점사이의 범위를 탐색 범위로서 결정하게 된다.
- <59> 상기 시작점과 상기 끝점에 의해 탐색 범위가 결정되면 420단계로 진행하여 상기 탐색 범위에 존재하는 코드워드들을 대상으로 하여 풀-탐색을 수행한다. 한편, 상기 풀-탐색에 의해 상기 탐색 구간 내에 존재하는 코드워드들 중 최소의 양자화 에러 값을 가

지는 최적의 코드워드가 결정되면 상기 코드워드에 대응한 인덱스를 최적의 인덱스로 설정한다. 그 후 상기 입력 벡터  $\vec{x}$ 에 대해 상기 설정된 최적의 인덱스에 대응하는 코드워드로서 양자화를 수행하게 된다.

<60> 전술한 본 발명의 실시 예에 있어 평균적인 시작점과 끝점간의 거리, 즉 평균 탐색 대상 코드-북의 크기  $N'$ 은 전체 코드-북의 크기  $N$ 보다 작다. 그러므로 풀-탐색에 있어 인코딩 복잡도와 관련된 탐색 코드-북의 크기를 줄임으로써 복잡도 문제를 개선시킬 수 있다.

<61> 도 6a는 본 발명의 실시 예를 설명하기 위한 코드-북의 일 예를 도시하고 있는 도면이며, 도 6b는 상기 도 6a의 코드-북을 재 정렬한 일 예를 도시하고 있는 도면이다.

<62> 이하 도 6a와 도 6b를 참조하여 본 발명의 실시 예에 따른 동작의 일 예를 설명하면 다음과 같다. 상기 도 6a에서 보이고 있는 코드워드들  $y_1, y_2, y_3, y_4, \dots, y_{14}, y_{15}$  각각의  $j$ 번째 요소 값들에 의해 내림차순으로 상기 코드워드들  $y_1, y_2, y_3, y_4, \dots, y_{14}, y_{15}$ 를 재 정렬한다. 상기 코드워드들  $y_1, y_2, y_3, y_4, \dots, y_{14}, y_{15}$ 를 내림차순에 의해 재 정렬하면 상기 도 6b와 같이  $y_1, y_3, y_4, y_2, y_7, y_5, y_6, y_9, y_{13}, y_{12}, y_{15}, y_8, y_{11}, y_{14}, y_{10}$ 의 순서에 의해 재 정렬된 코드-북이 생성된다. 상기 도 6b에서도 보여지고 있는 바와 같이 상기 생성된 코드-북은 상기  $j$ 번째 요소의 값의 크기 순서에 의해 내림차순으로 재배열되었음을 알 수 있다.

<63> 여기서 입력 벡터  $\vec{x}$ 의  $j$ 번째 요소 값이 '50'이라 가정하고, 실험에 의해 결정된 왜곡 값  $dist$ 와 제어 팩타  $P$ 를 이용하여 계산한 임계치  $thrs$ 를 100으로 가정한다.

- <64>      상기 가정하고 있는 입력 벡터  $\bar{x}$ 의 j번째 요소 값과 임계치  $\text{thrs}$ 를 상기 <수학식 7>에 적용함으로써 상위 경계 값은 '150'으로 결정되며, 하위 경계 값은 '-50'으로 결정된다.
- <65>      상기 코드-북의 재 정렬이 이루어지면 상기 상위 경계 값인 150과 상기 j 번째 요소 값들을 순차적으로 비교하여 상기 150보다 작은 값을 가지는 j 번째 요소 값을 찾는다. 상기 도 6b에서 보면 상기 150보다 작은 최초의 j 번째 요소 값은 149임을 알 수 있다. 따라서, 상기 149를 j 번째 요소 값으로 가지는 시점을 시작점으로 결정한다. 다음으로, 하위 경계 값인 -50과 상기 j 번째 요소 값들을 순차적으로 비교하여 상기 -50보다 작은 값을 가지는 j 번째 요소 값을 찾는다. 상기 도 6b에서 보면 상기 -50보다 작은 최초의 j 번째 요소 값은 -73임에 따라 상기 내림차순으로 정렬된 j 번째 요소 값들 중 상기 -73보다 큰 -47을 j 번째 요소 값으로 하는 시점을 끝점으로 결정한다. 따라서, 최종적인 탐색 범위는 상기 재 정렬된 코드워드들 중 149를 j 번째 요소 값으로 하는  $y_5$ 와 -47을 j 번째 요소 값으로 하는  $y_{11}$ 사이로 결정된다. 즉, 탐색 대상이 되는 코드워드들은  $y_5, y_6, y_9, y_{13}, y_{12}, y_{15}, y_8, y_{11}$ 로 결정된다. 그 후 최적의 코드워드를 찾기 위한 풀-탐색은 상기 입력 벡터  $\bar{x}$ 와 상기 탐색 대상이 되는 코드워드들 각각에 대해 이루어진다.
- <66>      전술한 바와 같은 본 발명은, 일반적으로 음성 및 영상 신호처리 분야에서 널리 사용되고 있는 NNVQ 알고리즘에 대하여, SNR 성능 감쇠가 없고 어떠한 추가적인 메모리도 요구하지 않는 고속 탐색 알고리즘을 제안하고 있다. 특히, 본 발명의 기술을 적용한 예로써,  $\rho=0.86$ 의 상관(correlation) 계수를 가지는 Gauss-Markov source에 대해 길이가 64이고, 차원이 3인 코드-북을 이용하여 벡터 양자화 수행 시, 코드-북의 평균 탐색 범

위는 약 52% 감소되었다. 또한, 본 발명의 기술을 적용한 다른 예로써, 3GPP의 비동기방식의 협대역 음성부호화기 표준인 AMR(Adaptive Multi-Rate) 음성 부호화기의 경우, 12.2kbps 모드에서 LSF(Line Spectral Frequency) 코드-북 양자화 시, SNR 성능 감쇠 없이 덧셈/뺄셈 연산은 평균 51.4%, 곱하기 연산은 평균 51.5%, 비교 연산은 평균 58.1%의 연산 감소효과를 보였으며, 약 60% 정도의 평균 탐색 코드-북의 크기를 줄일 수 있었다.

#### 【발명의 효과】

<67> 상술한 바와 같은 본 발명에서는, 코드-북의 코드워드들 중 입력 벡터에 대한 최적의 코드워드가 존재할 확률이 높은 탐색 범위 내의 코드워드들에 대해서만 풀-탐색을 적용함으로써 최소의 메모리를 사용하여 벡터 양자화를 위한 탐색 시간을 줄일 수 있는 효과를 가진다. 또한, 입력 벡터에 대한 벡터 양자화를 수행함에 있어 요구되는 연산을 줄임으로서 처리 절차를 간소화하는 효과를 가진다.

**【특허청구범위】****【청구항 1】**

입력 벡터를 벡터 양자화하기 위한 최적의 코드워드를 복수의 코드워드들로 구성된 코드-북으로부터 탐색하는 방법에 있어서,

상기 입력 벡터를 구성하는 복수의 요소들 중 임의 위치에 대응하는 요소 값에 미리 결정된 임계치를 감산 및 가산하여 하위 경계 값과 상위 경계 값을 계산하는 과정과,

상기 코드-북을 구성하는 상기 코드워드들 각각에 있어 상기 임의 위치에 대응하는 요소 값들 중 상기 하위 경계 값과 상기 상위 경계 값 사이에 존재하는 요소 값들을 가지는 코드워드들을 탐색 대상 코드워드들로 지정하는 과정과,

상기 탐색 대상 코드워드들을 대상으로 하여 상기 입력 벡터와의 양자화 에러 값을 계산하는 풀-탐색을 통해 최소 양자화 에러 값을 가지는 최적의 코드워드들 결정하는 과정을 포함함을 특징으로 하는 상기 방법.

**【청구항 2】**

제1항에 있어, 상기 임계치는 훈련 벡터들에 대한 MSE 값들 중 최대 값과, 상기 탐색 대상 코드워드들의 수를 최소화하도록 실험에 의해 결정한 제어 요소에 의해 미리 결정함을 특징으로 하는 상기 방법.



**【청구항 3】**

제2항에 있어서, 상기 제어 요소는 0과 1사이의 값으로 결정함을 특징으로 하는 상기 방법.

**【청구항 4】**

제1항에 있어서, 상기 탐색 대상 코드워드들을 지정하는 과정은,

상기 코드-북을 구성하는 상기 코드워드들 각각에 있어 상기 임의 위치에 대응하는 요소 값들의 크기에 의해 내림차순으로 상기 코드-북의 코드워드들을 재 정렬하는 과정과,

상기 상위 경계 값을 상기 재 정렬한 코드워드들 각각에 있어 상기 임의 위치에 대응하는 요소 값들과 비교하여 최초로 상기 상위 경계 값보다 작은 요소 값에 대응하는 시점을 시작점으로 결정하는 과정과,

상기 하위 경계 값을 상기 재 정렬한 코드워드들 각각에 있어 상기 임의 위치에 대응하는 요소 값들과 비교하여 최초로 상기 하위 경계 값보다 작은 요소 값에 대해 큰 요소 값에 대응하는 시점을 끝점으로 결정하는 과정과,

상기 시작점과 상기 끝점간에 존재하는 코드워드들을 상기 탐색 코드워드들로 지정하는 과정을 포함함을 특징으로 하는 상기 방법.

**【청구항 5】**

제1항에 있어서, 상기 탐색 대상 코드워드들을 지정하는 과정은,

상기 코드-북을 구성하는 상기 코드워드들 각각에 있어 상기 임의 위치에 대응하는 요소 값들의 크기에 의해 올림차순으로 상기 코드-북의 코드워드들을 재 정렬하는 과정과,

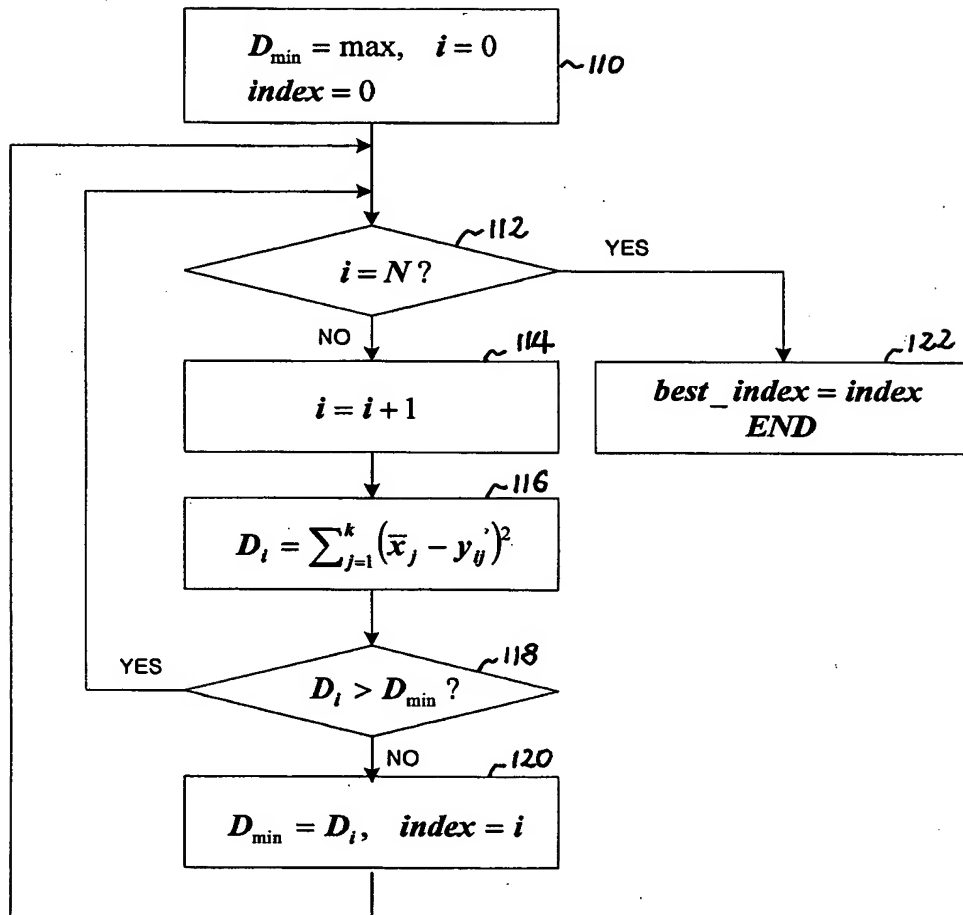
상기 하위 경계 값을 상기 재 정렬한 코드워드들 각각에 있어 상기 임의 위치에 대응하는 요소 값들과 비교하여 최초로 상기 하위 경계 값보다 큰 요소 값에 대응하는 시점을 시작점으로 결정하는 과정과,

상기 상위 경계 값을 상기 재 정렬한 코드워드들 각각에 있어 상기 임의 위치에 대응하는 요소 값들과 비교하여 최초로 상기 하위 경계 값보다 큰 요소 값에 대해 작은 요소 값에 대응하는 시점을 끝점으로 결정하는 과정과,

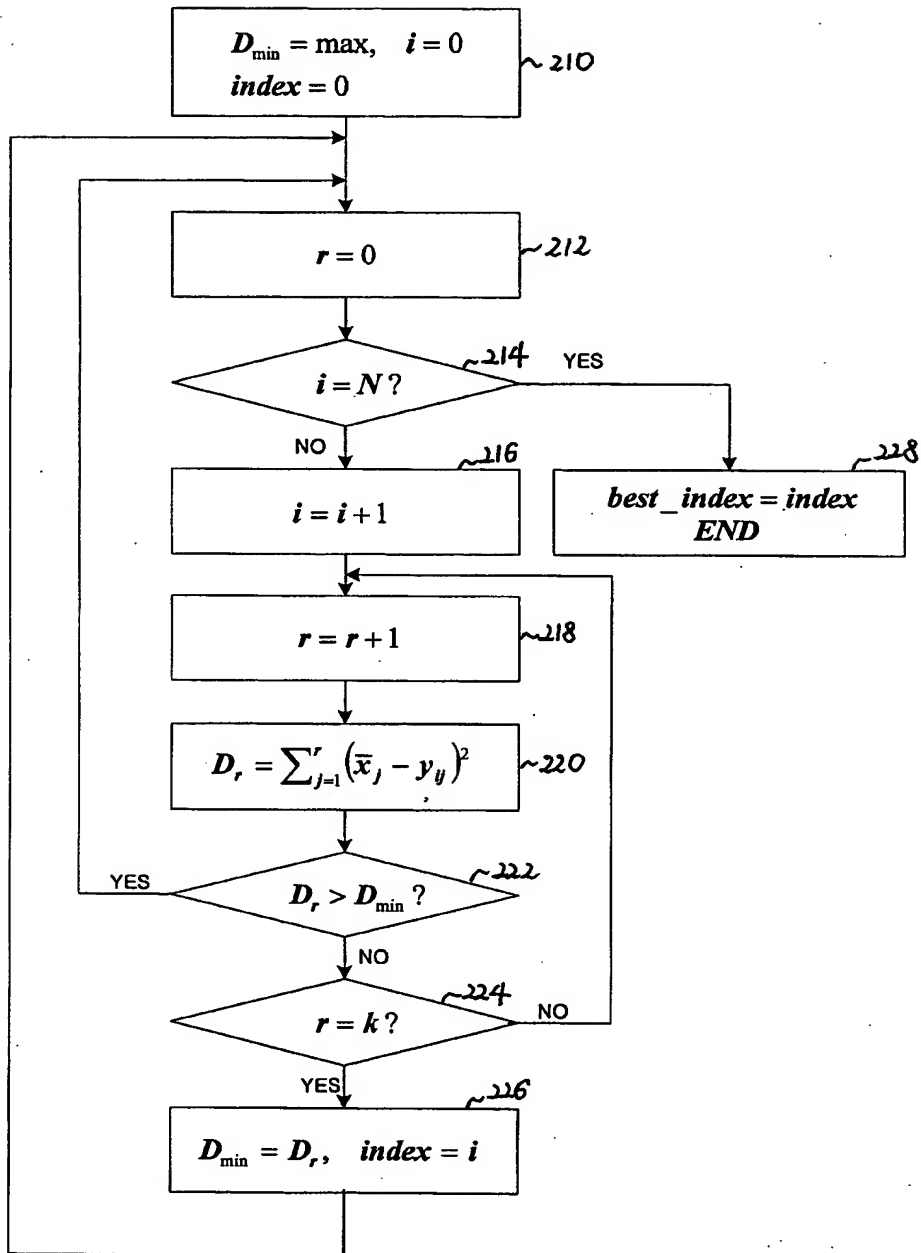
상기 시작점과 상기 끝점간에 존재하는 코드워드들을 상기 탐색 코드워드들로 지정하는 과정을 포함함을 특징으로 하는 상기 방법.

【도면】

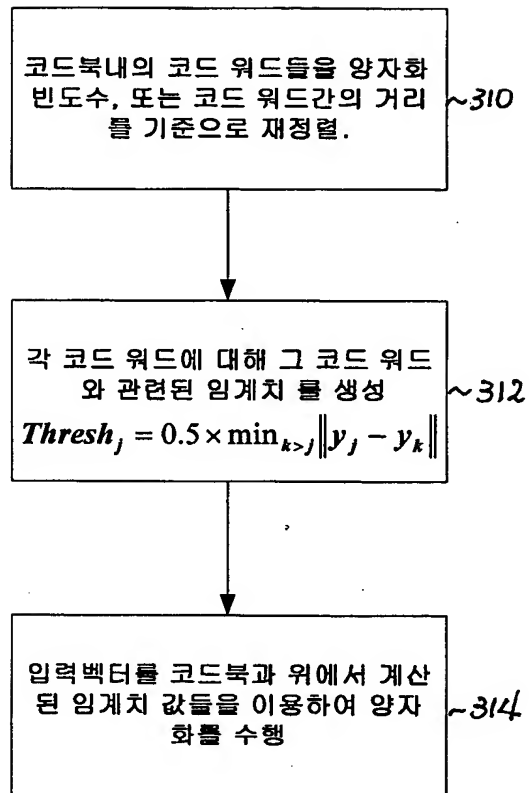
【도 1】



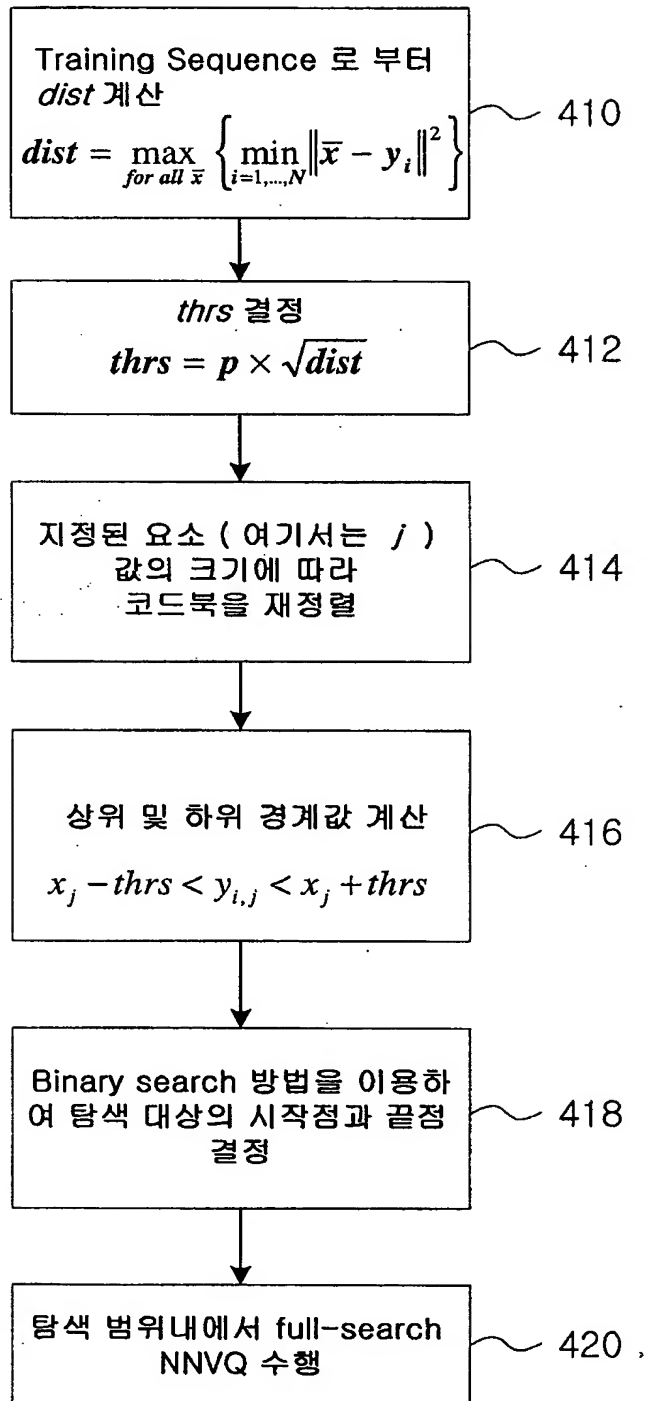
【도 2】



【도 3】



【도 4】



【도 5】

	1	2	3	.....	j	.....	k-1	k
$y_1$					$y_{1,j}$			
$y_2$					$y_{2,j}$			
$y_3$					$y_{3,j}$			
$y_4$					$y_{4,j}$			
.					.			
.					.			
.					.			
$y_i$					$y_{i,j}$			
.					.			
.					.			
.					.			
$y_{N-1}$					$y_{N-1,j}$			
$y_N$					$y_{N,j}$			

【도 6a】

	1	2	3	.....	j	.....	k-1	k
$y_1$					420			
$y_2$					270			
$y_3$					353			
$y_4$					310			
$y_5$					149			
$y_6$					80			
$y_7$					172			
$y_8$					-30			
$y_9$					68			
$y_{10}$					-90			
$y_{11}$					-47			
$y_{12}$					5			
$y_{13}$					43			
$y_{14}$					-73			
$y_{15}$					-9			

【도 6b】

	1	2	3	.....	j	.....	k-1	k
y <sub>1</sub>					420			
y <sub>3</sub>					353			
y <sub>4</sub>					310			
y <sub>2</sub>					270			
y <sub>7</sub>					172			
y <sub>5</sub>					149			
y <sub>6</sub>					80			
y <sub>9</sub>					68			
y <sub>13</sub>					43			
y <sub>12</sub>					5			
y <sub>15</sub>					-9			
y <sub>8</sub>					-30			
y <sub>11</sub>					-47			
y <sub>14</sub>					-73			
y <sub>10</sub>					-90			

← 시작점

검색범위

← 끝점